

Unconstrained Monotonic Calibration of Predictions in Deep Ranking Systems

USTC & Kuaishou Technology

Yimeng Bai, Shunyu Zhang, Yang Zhang, Hu Liu, Wentian Bao, Enyun Yu, Fuli Feng, Wenwu Ou

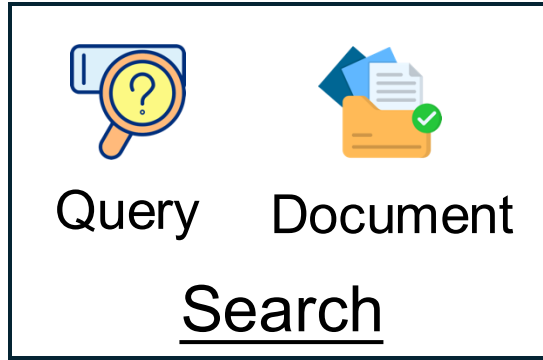
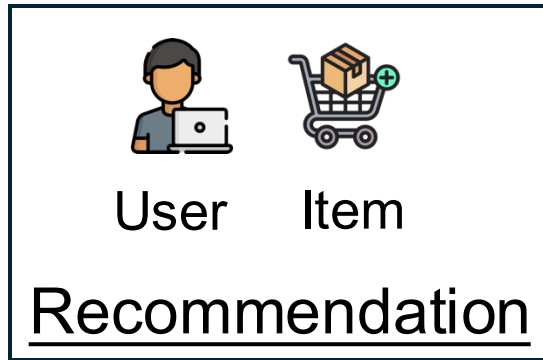
Outline

- Background
- Methodology
- Experiment
- Conclusion

Background

➤ What is Calibration

- ❑ **Ranking system** learns matching scores for sorting candidates



➡ Main focus: **relative order**
Larger score, larger similarity

- ❑ Accurate **absolute values** are essential for certain downstream tasks



Expected Revenue (ER) = predicted Click-Through Rate (pCTR) × price

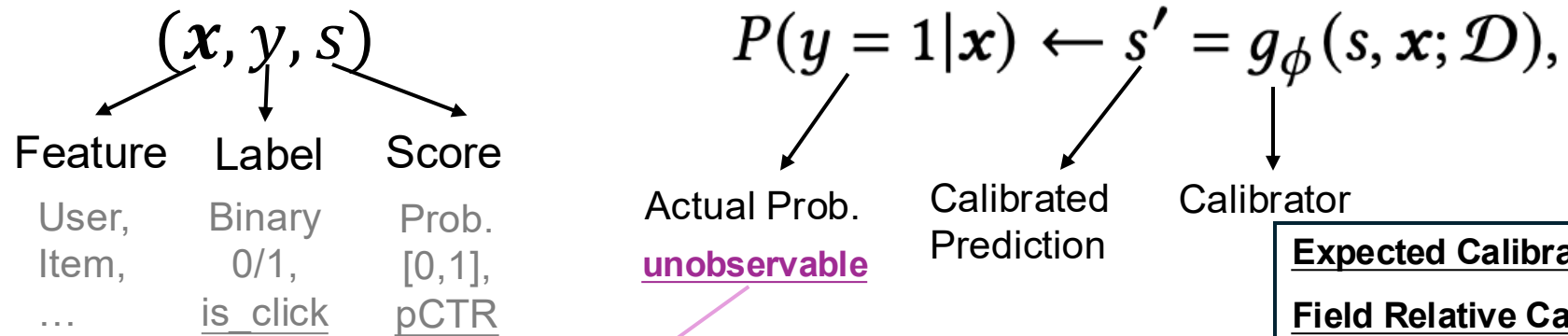
Assume price=10000, pCTR=0.8 → ER=8000
pCTR=0.9 → ER=9000

ΔpCTR=0.1 leads to ΔER=1000 !!!

Background

➤ How to Do Calibration

- Learn a **calibrator model** for adjustment



- Calibration metrics for evaluation

- We need the estimation of ground-truth $P(y=1|\mathbf{x})$

Aggregate samples with the same _____

- Score range ($pCTR \in [0.2, 0.4]$)
- Feature value (gender==man)

Expected Calibration Error (ECE)

Field Relative Calibration Error (FRCE)

Multi-Field Relative Calibration Error (MFRCE)

$$ECE = \frac{1}{|\mathcal{D}|} \sum_{k=1}^M \left| \sum_{(\mathbf{x}, y, s) \in \mathcal{D}} (s' - y) \cdot \mathbb{I}(s' \in [\frac{k-1}{M}, \frac{k}{M})) \right|,$$

$$FRCE = \frac{1}{|\mathcal{D}|} \sum_k \frac{|\sum_{(\mathbf{x}, y, s) \in \mathcal{D}} (s' - y) \cdot \mathbb{I}(z = z_k)|}{|\sum_{(\mathbf{x}, y, s) \in \mathcal{D}} y \cdot \mathbb{I}(z = z_k)|},$$

$$MFRCE = \frac{1}{N_z} \sum_z FRCE,$$

Background

➤ Existing Work on Calibration

❑ Common process: $s \in [0,1] \xrightarrow{\sigma^{-1}} l \in (-\infty, +\infty) \xrightarrow{\text{Key: calibrate}} l' \in (-\infty, +\infty) \xrightarrow{\sigma} s' \in [0,1]$

\downarrow \downarrow \downarrow \downarrow
Original score Original logit New logit New score

❑ Fixed-form **order-preserving** transformation

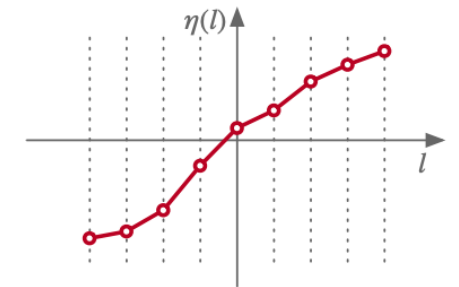
❑ **Simple transformation**

❑ Platt scaling: learn a logistic regression function $l' = \frac{1}{1 + e^{-(al+b)}}$

❑ Binning: Split samples into different bins based on l ; $l' = \text{Binning}(l)$

Statistically decide linear coefficients for each bins

❑ Limited parameter space



Isotonic Line-Plot Scaling

Background

➤ Existing Work on Calibration

□ DNN-based transformation

- FAC (WWW'20): add DNN bias $l' = \text{Binning}(l) + \text{DNN}(x)$
- SBCR (KDD'24): learn DNN-based linear coefficients $l' = \text{Binning}_{\text{DNN}}(l, x)$
- DESC (KDD'24): learn DNN-based weights of basis functions $l' = w_{\text{DNN}}(x) \cdot \mathcal{B}(l)$

□ Limitation

- Excessive constraints: Most adhere to piece-wise linear
 - Fail in complex scenarios like multi-field calibration
 - Amplify value errors in specific fields within feedback loops

- Motivation: **reduce constraints** on the architecture for full potential

$$\mathcal{B}_i^{\text{power}}(t; h_i) = x^{h_i}$$

$$\mathcal{B}_i^{\text{log}}(t; v_i) = \frac{\log(1 + v_i \cdot t)}{\log(1 + v_i)}$$

$$\mathcal{B}_i^{\text{scaling}}(t; a_i) = \sigma(\sigma^{-1}(t) \cdot a_i) ,$$

Methodology

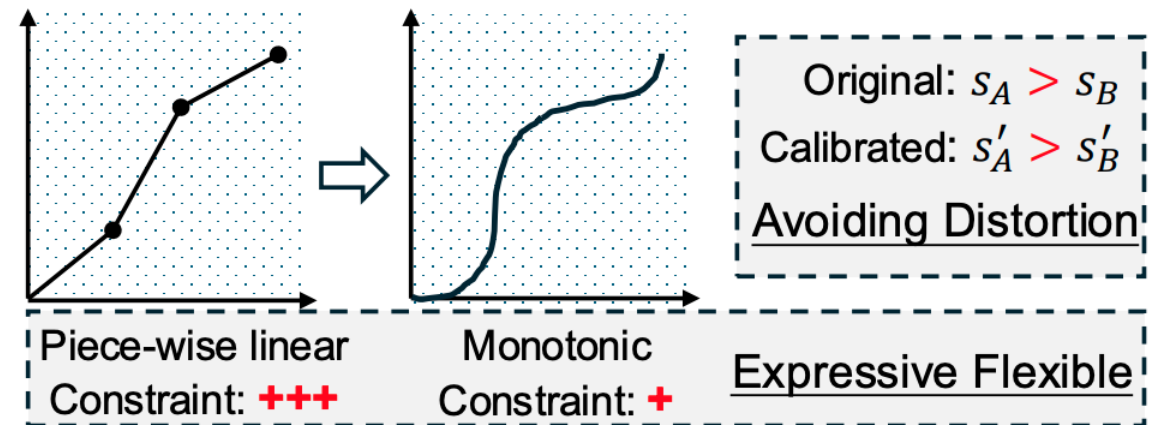
➤ How to Reduce Constraints

□ Core idea

- Piece-wise linear = linear + monotonic
- A more general form: only monotonic

□ How to design monotonicity

- **UMNN** [1] is the answer
- **Key**: if a function is differentiable, its derivative being **single-sign** is a necessary and sufficient condition for the function to be **monotonic**
- **How**: construct a single-sign function and do integral computation



Methodology

➤ Overall Framework

□ Monotonic Calibrator Architecture

□ **UMNN** to ensure monotonicity

□ Rescaling based on features

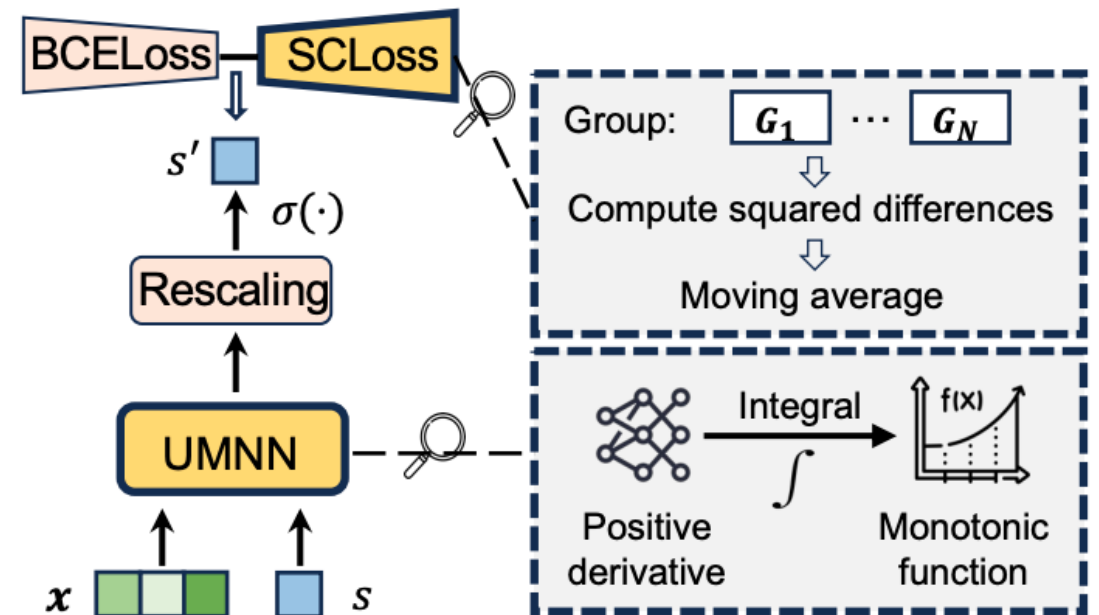
□ Calibration-aware Learning Strategy

□ BCELoss

□ Common setting in previous work

□ **SCLoss** (Smooth Calibration Loss)

□ Specifically designed to emphasize calibration aspect



Methodology

➤ Monotonic Calibrator Architecture

□ UMNN

- MLP with **positive outputs**

$$h(t, \mathbf{x}) = 1 + \text{ELU}(\text{MLP}([t; \mathbf{x}])),$$

- Compute **integral** based on specific algorithm

$$U(s, \mathbf{x}) = \int_{t=0}^{\boxed{t=\sigma^{-1}(s)}} h(t, \mathbf{x}) dt + \beta.$$

Original logit

- Monotonicity is ensured by $\frac{d}{ds} U(s, \mathbf{x}) = h(\sigma^{-1}(s), \mathbf{x}) \cdot (\sigma^{-1}(s))' > 0,$

□ Rescaling

- Features \mathbf{x} are fed into a distinct MLP module

$$g_{\phi}(s, \mathbf{x}) = \sigma(e^{w(\mathbf{x})} \cdot U(s, \mathbf{x}) + b(\mathbf{x})).$$

- Compute a feature-specific rescaling factor $w(\mathbf{x})$ and bias term $b(\mathbf{x})$
- Enhance the overall network learnability and preserve monotonicity
- Apply Sigmoid to transform logit to score

Methodology

➤ Calibration-aware Learning Strategy

□ SCLoss

□ Computation

- Discretize the interval into N equal-width bins
- Group samples by bins calibrated predictions fall into
- Compute the squared differences per bin
- Take a weighted average based on the group size

□ Moving average technique

- Smooth by keeping a fraction of values
from the previous batch

$$\text{SCLoss}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{k=1}^N |G_k| (\bar{y}_k - \bar{s}'_k)^2,$$
$$G_k = \{(\mathbf{x}, y, s) \in \mathcal{B} \mid \frac{k-1}{N} \leq s' < \frac{k}{N}\},$$

$$\bar{y}_k \leftarrow \tau \cdot \bar{y}_k + (1 - \tau) \cdot \frac{1}{|G_k|} \sum_{(\mathbf{x}, y, s) \in G_k} y,$$
$$\bar{s}'_k \leftarrow \tau \cdot \bar{s}'_k + (1 - \tau) \cdot \frac{1}{|G_k|} \sum_{(\mathbf{x}, y, s) \in G_k} s',$$

□ Overall Loss

$$\mathcal{L}(\mathcal{B}) = \text{BCELoss}(\mathcal{B}) + \lambda \cdot \text{SCLoss}(\mathcal{B}),$$

Experiment

➤ Offline Experiment Setting

□ Dataset

- CTR dataset: Avazu, AliCCP

□ Evaluation

- Ranking: AUC, GAUC
- Calibration (main focus): ECE, FRCE, MFRCE

□ Baseline

- Simple: Uncalib, HistBin, IsoReg, SIR, Platt, Gauss
- DNN-based: FAC, SBCR, DESC

□ Pipeline

- Train a DeepFM model, then train a calibrator model to calibrate predictions

Experiment

➤ Offline Experiment Result

Method	Avazu					AliCCP				
	AUC↑	GAUC↑	ECE↓	FRCE↓	MFRCE↓	AUC↑	GAUC↑	ECE↓	FRCE↓	MFRCE↓
Uncalib	0.7413	0.6544	0.0413	0.4207	0.3484	0.6348	0.5782	0.0202	1.4824	0.7291
HistBin [41]	0.7309	0.6368	0.0090	0.2701	0.2035	0.5239	0.5068	0.0076	1.2112	0.3909
IsoReg [42]	0.7413	0.6544	0.0088	0.2313	0.1655	0.6348	0.5782	0.0078	1.0168	0.3555
SIR [13]	0.7413	0.6544	0.0083	0.2147	0.1478	0.6348	0.5782	0.0080	1.0751	0.3721
Platt [33]	0.7413	0.6544	0.0109	0.2294	0.1606	0.6348	0.5782	0.0078	1.0179	0.3560
Gauss [26]	0.7413	0.6544	0.0095	0.2346	0.1640	0.6348	0.5782	0.0078	1.0180	0.3561
FAC [30]	0.7515	0.6649	<u>0.0042</u>	0.1473	<u>0.0974</u>	0.6358	0.5791	0.0065	0.9758	0.3186
SBCR [44]	<u>0.7516</u>	<u>0.6651</u>	0.0051	<u>0.1416</u>	0.0994	0.6362	0.5797	<u>0.0063</u>	<u>0.9536</u>	<u>0.3096</u>
DESC [40]	0.7514	0.6637	0.0074	0.1549	0.1086	<u>0.6364</u>	0.5802	0.0067	0.9697	0.3200
UMC (ours)	0.7521	0.6654	0.0033	0.1172	0.0837	0.6367	<u>0.5801</u>	0.0058	0.9422	0.2951

1. **Calib > Uncalib**: calibration is necessary for ranking models
2. **DNN-based > Simple**: enhanced model capacity
3. **UMC > baselines**: expressive calibrator & calibration-aware learning

Experiment

➤ Ablation Experiment Result

Table 3: Results of the ablation study for UMC on Avazu.

Method	AUC↑	GAUC↑	ECE↓	FRCE↓	MFRCE↓
Full UMNN	0.7521	0.6654	0.0033	0.1172	0.0837
w/o Rescaling	0.7508	0.6630	0.0078	0.1558	0.1118
w/o Feature	0.7413	0.6544	0.0080	0.2119	0.1465
w/o UMNN	0.7413	0.6544	0.0095	0.2482	0.1792
Full SCLoss	0.7521	0.6654	0.0033	0.1172	0.0837
w/o Average	0.7508	0.6626	0.0084	0.1613	0.1248
w/o SCLoss	0.7519	0.6651	0.0072	0.1672	0.1154
w/ MSELoss	0.7521	0.6649	0.0061	0.1512	0.0982

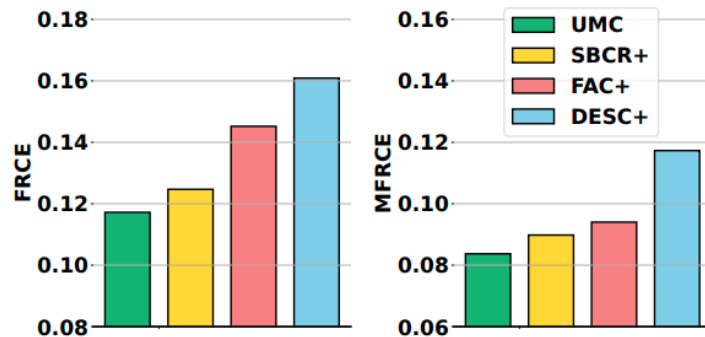


Figure 3: Results of adding SCLoss to other neural network-based baseline methods on Avazu.

- **UMC w/o Rescaling.** This variant removes the rescaling layer of the architecture and relies solely on the UMNN output.
- **UMC w/o Feature.** This variant eliminates the feature input of the UMNN, resulting in a univariate mapping function.
- **UMC w/o UMNN.** This variant replaces the UMNN with a piece-wise linear model in FAC.
- **UMC w/o Average.** This variant sets decay rate τ to zero, removing the exponential moving average usage of SCLoss.
- **UMC w/o SCLoss.** This variant sets balancing weight λ to zero, completely disables the effect of SCLoss.
- **UMC w/ MSELoss.** This variant replaces SCLoss with MSELoss, aligning the calibrated output with the binary label.

1. Designs of UMNN and SCLoss are all effective
2. SCLoss can be generally applied to baselines

Experiment

➤ Hyper-parameter Experiment Result

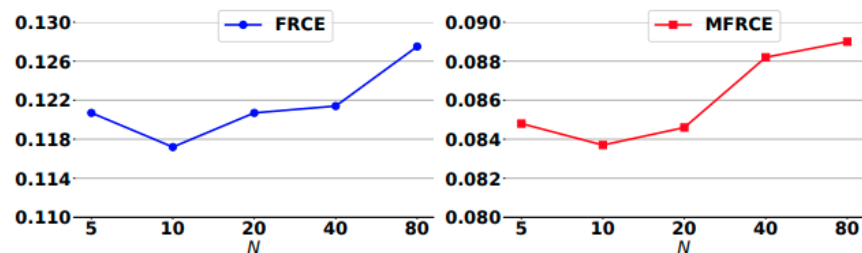


Figure 4: Results of the performance of UMC across different values of group number N on Avazu.

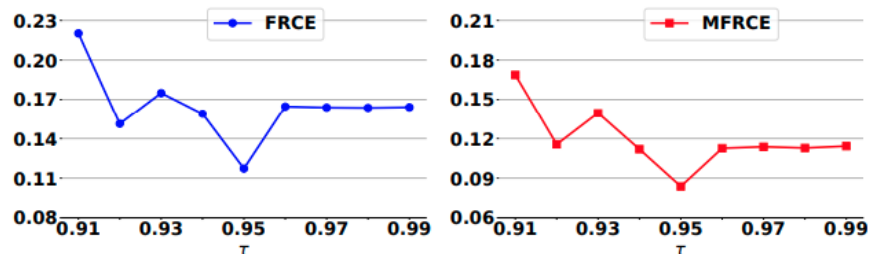


Figure 5: Results of the performance of UMC across different values of decay rate τ on Avazu.

$$\text{SCLoss}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{k=1}^N |G_k| (\bar{y}_k - \bar{s}'_k)^2,$$
$$G_k = \{(x, y, s) \in \mathcal{B} \mid \frac{k-1}{N} \leq s' < \frac{k}{N}\},$$

Group number N in SC Loss

1. **Insufficient numbers** diminishes the ability to differentiate among groups
2. **Excessive numbers** can lead to bins with sparse data, rendering the computation unreliable

Decay rate τ in SC Loss

1. **Too small values** lead to a delayed response to recent batch data
2. **Too large value** obscure long-term trends

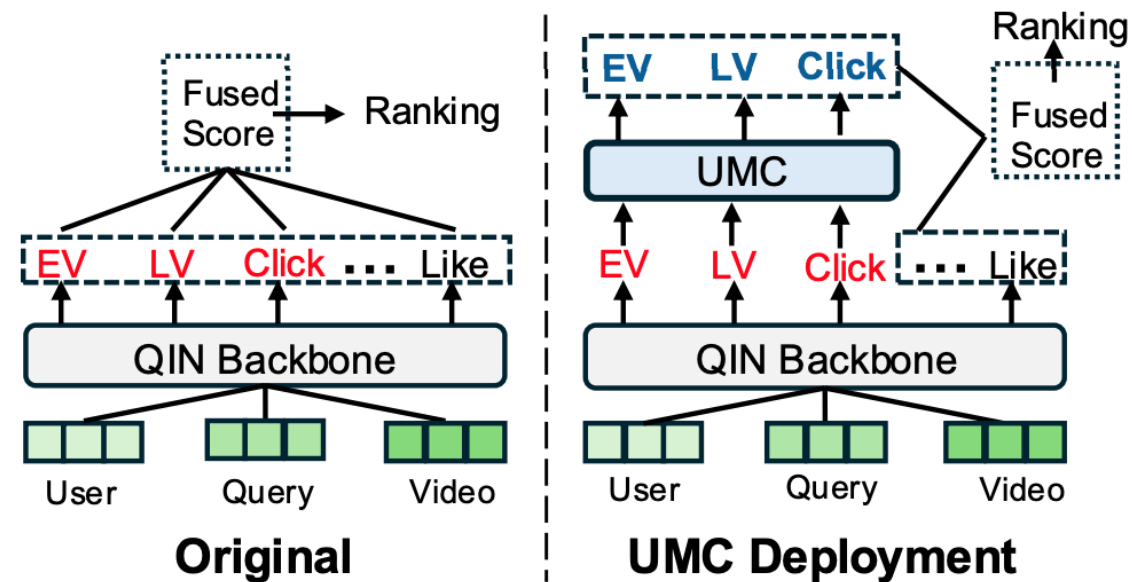
$$\bar{y}_k \leftarrow \tau \cdot \bar{y}_k + (1 - \tau) \cdot \frac{1}{|G_k|} \sum_{(x, y, s) \in G_k} y,$$
$$\bar{s}'_k \leftarrow \tau \cdot \bar{s}'_k + (1 - \tau) \cdot \frac{1}{|G_k|} \sum_{(x, y, s) \in G_k} s',$$

Experiment

➤ Online Experiment Result

❑ Kuaishou Video Search System

- ❑ Backbone: QIN [2]
- ❑ Scale: 5% (20 million users)
- ❑ Time: one week
- ❑ Method: **replace** the original predictions of EffectiveView, LongView, Click in the fused score
- ❑ Metrics: total watch time (WT) effective video views (VV) and click-through rate (CTR)
- ❑ Note: **efficient** deployment without introducing significant computational overhead



Metric	WT	VV	CTR
Improvement	+0.414%	+0.411%	+0.170%

Conclusion

➤ Calibration in Ranking System

- ❑ **Challenge:** excessive calibrator constraints
- ❑ **Motivation:** reduce constraints for a more general form
- ❑ **Methodology**
 - ❑ Monotonic Calibrator Architecture (UMNN)
 - ❑ Calibration-aware Learning Strategy (SCLoss)
- ❑ **Offline** experiment on real-world datasets
- ❑ **Online** experiments on Kuaishou video search platform

Thanks



The code has been released at
<https://github.com/baiyimeng/UMC>